

Fully Automatic Expression-Invariant Face Correspondence

Augusto Salazar^{*†} Stefanie Wuhler^{†‡} Chang Shu[‡] Flavio Prieto[§]

February 8, 2012

Abstract

We consider the problem of computing accurate point-to-point correspondences among a set of human face scans with varying expressions. Our fully automatic approach does not require any manually placed markers on the scan. Instead, the approach learns the locations of a set of landmarks present in a database and uses this knowledge to automatically predict the locations of these landmarks on a newly available scan. The predicted landmarks are then used to compute point-to-point correspondences between a template model and the newly available scan. To accurately fit the expression of the template to the expression of the scan, we use as template a blendshape model. Our algorithm was tested on a database of human faces of different ethnic groups with strongly varying expressions. Experimental results show that the obtained point-to-point correspondence is both highly accurate and consistent for most of the tested 3D face models.

1 Introduction

We consider the problem of computing point-to-point correspondences among a set of human face scans with varying expressions in a fully automatic way. This problem arises from building a statistical model that encodes face shape and expression simultaneously using a database of human face scans. In order to build a statistical model, we rely on the correct computation of dense point-to-point correspondences among the subjects of a database. That is, the raw scans have to be parameterized in such a way that likewise anatomical parts correspond across the models [1]. Facial expression affects the geometry of the human face and therefore is important for facial shape analysis. A statistical model of face shapes and expressions can be used in applications such as face recognition, expression recognition, or reconstructing accurate 3D models of faces from input images [2, 3, 4, 5, 6].

^{*}Perception and Intelligent Control research group, National University of Colombia.

[†]Cluster of Excellence, Multimodal Computing and Interaction, Saarland University.

[‡]National Research Council of Canada.

[§]GAUNAL research group, National University of Colombia.

Computing accurate point-to-point correspondences for a set of face shapes in varying expressions is a challenging task because the face shape varies across the database and each subject has its own way to perform facial expressions. The problem is further complicated by incomplete and noisy data in the scans.

While many approaches have been proposed to compute point-to-point correspondences [7], only few of them have been applied to statistical model building and shape analysis of human face shapes. Blanz and Vetter [2], built a statistical model called morphable model for a set of 3D face scans with varying expressions. The correspondence algorithm is based on using optical flow on the texture information of the faces. This assumes that the faces are approximately spatially aligned. Xi and Shu [8] built a statistical model based on principal component analysis for a set of 3D face scans with neutral expressions. The correspondence algorithm is based on fitting a template model to the scans using a non-rigid iterative closest point algorithm. To start this algorithm, the faces need to be approximately aligned using a set of manually placed marker positions. Both of these registration approaches fail for misaligned models.

In this work, we develop a novel technique to compute correspondences between a set of facial scans with varying expressions that does not require the scans to be spatially aligned. Our correspondence computation procedure uses a template model P as prior knowledge on the geometry of the face shapes. Unlike Xi and Shu [8], we aim to find correspondences for faces with varying expressions. Hence, it is not enough to have a template model that captures the face shape of a generic model, but we also need to capture the expressions of a generic model. To achieve this, we model P as a blendshape model as in Li et al. [9]. In a blendshape model, expressions are modeled as a linear combination of a set of basic expressions. Hence, blendshape models are both simple and effective to model facial expressions.

Our approach proceeds as follows. We first use a database of human face scans with manually placed landmark positions to learn local properties and spatial relationships between the landmarks using a Markov network. Given an input scan F without manually placed landmarks, we first predict the landmark positions on F by carrying out statistical inference over the trained Markov network. Section 3 discusses this step. In order to perform statistical inference, we need to restrict the search region for each landmark. This is detailed in Section 4. The predicted landmarks are used to align P to F . In order to fit the expression of P to the expression of F , the weights of the generic blendshape model are optimized as discussed in Section 5.1. Finally, the shape of P is changed to fit the shape of F as outlined in Section 5.2. Fig. 1 shows an overview of the method.

2 Related Work

This section reviews literature in face shape analysis related to finding landmarks on face models, computing correspondences between three-dimensional shapes, and using blendshape models for facial animation.

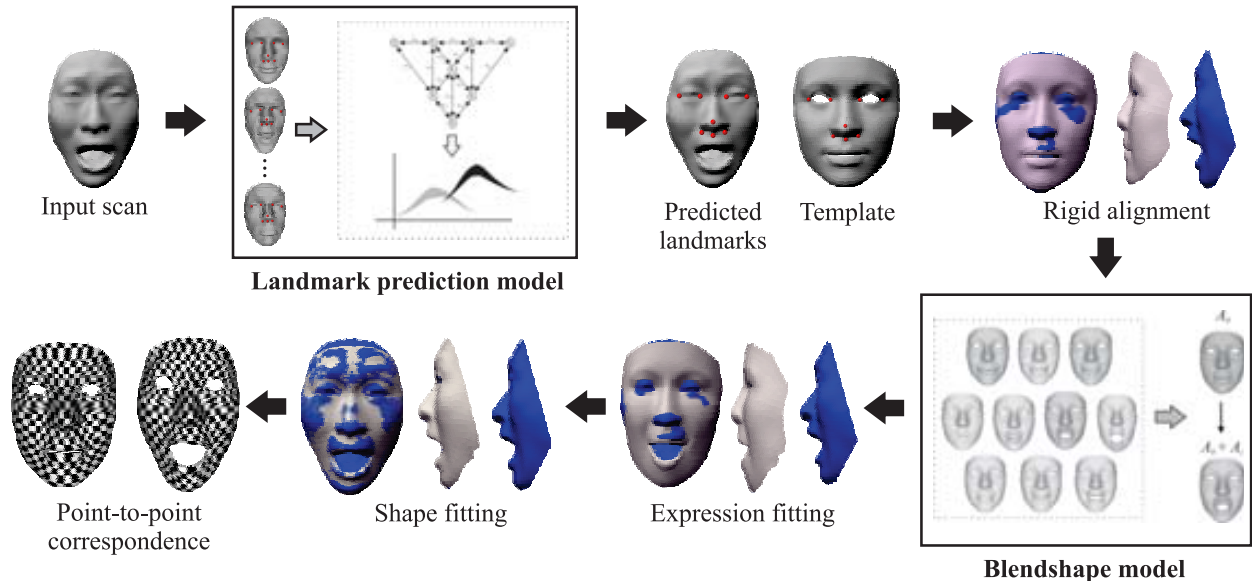


Figure 1: Overview of the fully automatic expression-invariant face correspondence approach.

2.1 Finding Landmarks on Face Models

Traditionally, facial feature detection is done in 2D images, but recent developments on 3D data acquisition have allowed to overcome the problems attached to the 3D technologies. Existing registration methods demonstrated that landmark-based methods provide more accurate and consistent results. However, only a few approaches consider 3D landmark detection, while accounting for expression and pose variations [10].

Ben Azouz et al. [11] propose a method to find correspondences by automatically predicting marker positions on 3D models of a human body. The method encodes the statistics of a surface descriptor and geometric properties at the locations of manually placed landmarks in a Markov network. This method works only for models with slight variation of posture. Mehryar et al. [10] introduce an algorithm to automatically detect eyes, nose, and mouth on 3D faces. The algorithm correctly detects the landmarks in the presence of pose, facial expression and occlusion variations. This method is useful as initial alignment but not for an accurate registration. Creusot et al. [12] present a method to localize a set of 13 facial landmark points under large pose variation or when occlusion is present. Their method learns the properties of a set of descriptors computed at the landmark locations and encodes both local information and spatial relationships into a graph. The method works well for neutral pose. However, in the presence of expression variation, the accuracy decreases considerably.

As our aim is to obtain accurate point-to-point correspondences, we derived a landmark prediction method based on the approach of Ben Azouz et al. [11]. The surface descriptor we used is able to catch the local geometry properly [13] and, by combining it with a canonical representation [14], our approach is able to detect landmarks in the presence of facial expressions. We select a machine learning-based approach to avoid classic assumptions about initial alignments of the scans and using local descriptor extrema as stable feature points.

The advantage is that learning-based approaches can easily be extended to other contexts.

2.2 Correspondence Computation

Several methods have been proposed to solve the problem of establishing a meaningful correspondence between shapes. Here, we focus on computing correspondences between human face shapes. Methods that do not assume templates usually have the problem that some points are not registered accurately. To remedy this, we assume a template model. In the following, we only review approaches that use template models (for details about methods for correspondence computation see the survey of van Kaick et al. [7]).

Passalis et al. [15] proposed a 3D face recognition method that uses facial symmetry to handle pose variation and missing data. A template is fitted to the shape of the input model as follows: an Annotated Face Model (AFM) [16] is iteratively deformed towards the input using automatically predicted landmarks and an algorithm based on Simulated Annealing. When dealing with facial expressions, the performance of the recognition system decreases. This is due to an incorrect registration of the mouth region. The authors do not show extensive evaluations of this fully-automatic registration method as this is not the main part of their work.

Statistical learning-based approaches have been effectively used to model facial variations oriented to both the synthesis and recognition of faces. Blanz and Vetter [2] developed a 3D morphable model (3DMM) for the synthesis of 3D faces from photographs. As the registration is specific to the scanning setup, rigid alignment of the scans is assumed. Lu and Jain [17] present an approach to perform face recognition using 3D face scans. The approach builds a 3DMM for each subject in the database. When a test image becomes available, the approach matches the scan to a specific individual using the learned 3DMM. Unlike our method, their training data is parameterized using manually placed landmarks and the test scans are parameterized using individual-specific deformation models. Basso et al. [18] extend the method of Blanz and Vetter [2] to register 3D scans of faces with arbitrary identity and expression. The rigid alignment of the scans is also assumed for registration. To avoid the use of texture information, Amberg et al. [19] present a method to fit a 3DMM to 3D face scans using only shape information. They demonstrate the performance of the method in the presence of expression variation, occlusion and missing data, but do not conduct extensive evaluations of the registration.

Registration methods based on iteratively deforming a template to the data are an alternative to statistical learning-based approaches. Allen et al. [20] present an approach to parameterize a set of 3D scans of human body shapes in similar posture. To fit the template to each scan, the method proceeds by using a non-rigid iterative closest point (ICP) framework coupled with a set of manually placed marker positions. Xi and Shu [8] extend the method of Allen et al. [20] to deform a template model to a head scan. The shape fitting is carried out as in Allen et al. [20] but uses radial basis functions to speed up the deformation process. Unlike our method, this only allows for neutral expressions and uses manually placed markers to align the template to a head scan. Wuhler et al. [21] propose a method to deform a template model to a human body scan in arbitrary posture. The

method works in two stages: posture and shape fitting. Posture fitting relies on the location of different landmarks, which are predicted in a fully automatic way using a statistical model of landmark positions learned from a population. Our method can be viewed as an extension of this approach, but instead of fitting the posture, we fit the expression using blendshapes (see Section 2.3).

Methods that compute a correspondence between two surfaces by embedding the intrinsic geometry of one surface into the other one by using Generalized Multi-Dimensional Scaling (GMDS) [22] are another alternative to deal with variations due to facial expressions [23]. The performance of these methods has been demonstrated for face recognition. As GMDS methods do not take care that close-by points on one surface map to close-by points on the other, the results are often spatially inconsistent. This prevents such methods from being used for shape analysis.

2.3 Use of Blendshape Models

Modeling expressions using blendshape models is an alternative to approaches based on statistical models where a comprehensive database annotation process has to be carried out to extract variational information. In a blendshape model, movements of the different facial regions are assumed to be independent. Any expression is then modeled as a linear combination of the differences between a set of basic expressions, called *blendshapes*, and a neutral expression. That is, to produce an expression, the displacements causing the movement are linearly combined. Using a representative set of blendshapes, this simple model is effective to model facial expressions.

Li et al. [9] propose a method to transfer the expression of a subject to an animated character. Their framework allows to create optimal blendshapes from a set of example poses of a digital face model automatically. To fit the expression of the subject to the character, a blendshape optimization is carried out in gradient space, where an optimal blending weight is estimated for every template vertex and expression. Weise et al. [24] present a framework for real-time 3D facial animation. The method tracks the rigid and non-rigid motion of the user’s face accurately. They incorporate the expression transfer approach of Li et al. [9] in order to find much of the variation from the example expressions. The registration stage requires offline training where a generic template is fitted to the face of a specific subject. To obtain the results, manual marking of features has to be carried out.

Because of the advantages of modeling expression using linear blendshapes, we use it to aid the shape matching. We only optimize a blending weight per expression. This reduces the dimension of the optimization space drastically. Since our database of blendshapes is small, the expression fitting stage of our algorithm is efficient and helps to improve the results significantly.

3 Landmark Prediction

This section outlines how to predict a set of landmark positions on a face scan. To establish the correspondences across the whole database, we fit a template to each model. The fitting process begins with the extraction of the locations of eight landmarks shown as red spheres in Fig. 2. The locations of the landmarks were selected based on the fact that in the presence of facial expressions, the corners of the eyes, and the base and tip of the nose do not move drastically. Each landmark is located automatically on the face surface by means of a Markov network following the procedure proposed by Ben Azouz et al [11]. The network learns the statistics of a property of the surface around each landmark and the structure of the connections shown in Fig. 2.

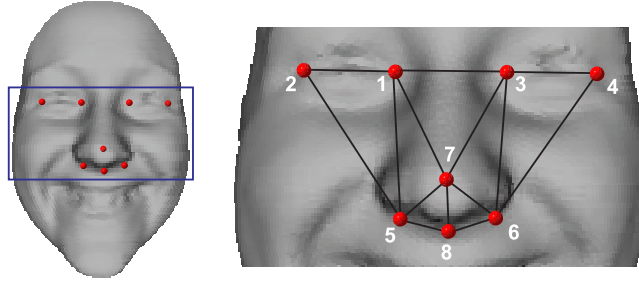


Figure 2: Face model with landmarks. Locations and landmark graph structure.

3.1 Learning

Two important aspects have to be defined for the training of the Markov network. First, each landmark l_i ($i = 1, 2, \dots, L$), represented by a network node, is described using a node potential. We use a surface descriptor called *Finger Print* (*FP*) [13], which is a measure related to the area of a geodesic circle centered at the point to be characterized. The descriptor at a point p_k ($k = 1, 2, \dots, N$, N is the number of vertices in the model) is obtained by computing the distortion of the geodesic disks with respect to their corresponding Euclidean disks. The final surface descriptor is a vector of distortions obtained by varying the radius of the geodesic disk (see Fig. 3). The reason we use *FP* as node potential is because it is isometry-invariant. Hence, in scenarios where the surface undergoes changes that preserve isometry, *FP* has been effective to encode the surface information of an object. *FP* is used to predict landmarks on human models in varying poses [25].

Second, a link between landmarks, represented by a network edge, is described using an edge potential. Although we selected the locations of the landmarks based on the observations that nose and eye regions do not change much in the presence of expressions, some distortions along the edges of the Markov network, may occur. To minimize the effects of the face movements, we compute the canonical form [14] of each model and define the edge potential as the relative position of landmark l_i with respect to landmark l_j in the canonical form space. We compute the canonical form as the embedding of the intrinsic geometry of the

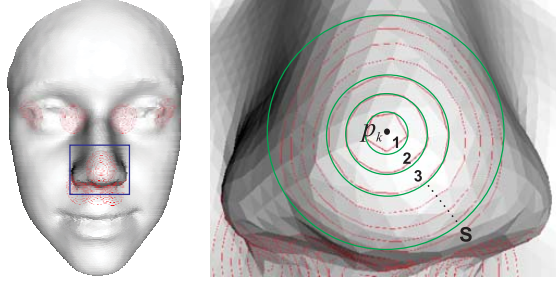


Figure 3: Circles used to compute the Finger Print descriptor. Red and green circles correspond to the Geodesic and Euclidean circles, respectively.

face surface to \mathbb{R}^3 . To compute this embedding, we perform least-squares multi-dimensional scaling [26] with geodesic distances between vertices as dissimilarities, and the geodesic distances are computed using fast marching [14]. We choose these standard techniques as they are efficient. These potentials ensure that the model is isometry-invariant.

The Markov network training process learns the distributions of both node and edge potentials. We assume Gaussian distributions for both the node and edge descriptors in this paper, and we learn the distributions using maximum likelihood estimation. We choose this distribution based on experimental observations.

3.2 Prediction

The Estimation of the location of landmarks on a test model is carried out by using probabilistic inference over the Markov network. In practice, we perform inference using the loopy belief propagation algorithm [27]. This algorithm requires a set of possible labels for each node. In our case, this means we need to provide a number of candidate locations for each landmark.

Wuhrer et. al [21] use the canonical forms to learn the average locations of the landmarks, but because of the flipping-invariant property of the canonical forms, it is necessary to compute eight different alignments and select the one that leads to the minimum distance between the scan and the deformed template. In this work, we design a method to restrict the search space based on a rough template alignment. In this way, only one fitting process has to be computed, reducing the computing cost by a factor of 8.

4 Restricting the search region

There are two reasons to reduce the search space for the landmarks: to increase the efficiency of the landmark prediction and to eliminate the ambiguity caused by the facial symmetry. We treat the problem of restricting the search region for the landmarks as a 3D face pose estimation problem. In our case, the estimated pose does not have to be so accurate since the Markov network refines the position of the landmarks, but it has to be accurate enough to identify the left and right sides of the face. The proposed face pose estimation method

finds four landmarks located on the nose region and extracts the information of the face symmetry planes by using a template of the landmark graph. Once the nose landmarks are labeled, the final position of the entire set of landmarks is obtained by transforming the template to the coordinate system of the test model. Fig. 4 shows the main steps of the proposed search space restriction method.

Before explaining the rough template alignment procedure, we introduce a method to classify a vertex of a 3D model into a specific class. In our case, the classes correspond to the nodes of the Markov network and the 3D model corresponds to a 3D face model. The decision rules are derived from a clustering procedure over the Principal Components Analysis (PCA) projections of a surface feature and a pre-selection method based on the surface primitives.

As the value of the FP descriptor at each landmark l_i was computed during the Markov network training process, we can model the distributions of the surface descriptors and use them to classify a vertex v_k on the face surface into a class i (each landmark corresponds to a class). PCA is a useful tool to compress a high-dimensional space into a linear low-dimensional space. When the space corresponds to a multidimensional feature space, sometimes, depending on the distinctiveness of the features, it is possible that elements of the same class form clusters in the PCA space. In our case, the FP descriptor can be viewed as S -dimensional vector and PCA is used to reduce the dimensionality to D . Fig. 5 shows the results of applying PCA to the data with neutral expression (for information about the database, see Section 6.1).

In PCA space, samples of the same class tend to form groups, which are slightly separate. As the eye corners and nose base landmarks are symmetric, there are six groups and some groups overlap. Since the data forms clusters, it is possible to define rules in order to assign a new sample into a specific class. We define a new cluster, denoted as M -cluster, by removing the samples which are farther than M ($M \in \mathbb{R}^+$) times the standard deviation from the cluster medoid. Medoids are representative objects of a cluster whose average dissimilarity to all the objects in the cluster is minimal [28]. For instance, Fig. 5 shows the M -clusters formed by setting $M = 1$. While some M -clusters are partially overlapping, we can see that the separation between classes improved over the initial.

We derive a rule E_i for a class i based on the clustering procedure. The rule E_i is defined as the minimum volume enclosing ellipsoid of a M -cluster $_i$ (see Fig. 5). E_i is obtained from the representation of the ellipsoid in the center form as $(p_k - C)^T A (p_k - C) \leq 1$, where C corresponds to the center of the ellipsoid and A is the 3×3 matrix of the ellipse equation. When a new point p_k becomes available, each E_i is evaluated in order to see if the point satisfies the equation. As some M -clusters are overlapping, it is possible that two or more labels are assigned to the same p_k . Similarly, it is possible that p_k is not assigned to any class because the point lies in a region that is not of interest. Fig. 6 shows an example of the vertex classification results obtained using the proposed method.

It is not efficient to compute the descriptor value and its projection to PCA space for all the vertices of the mesh. To reduce the search space, we compute samples on the surface using a curvature-based descriptor. More precisely, we use as samples all surface *umbilics* [29],

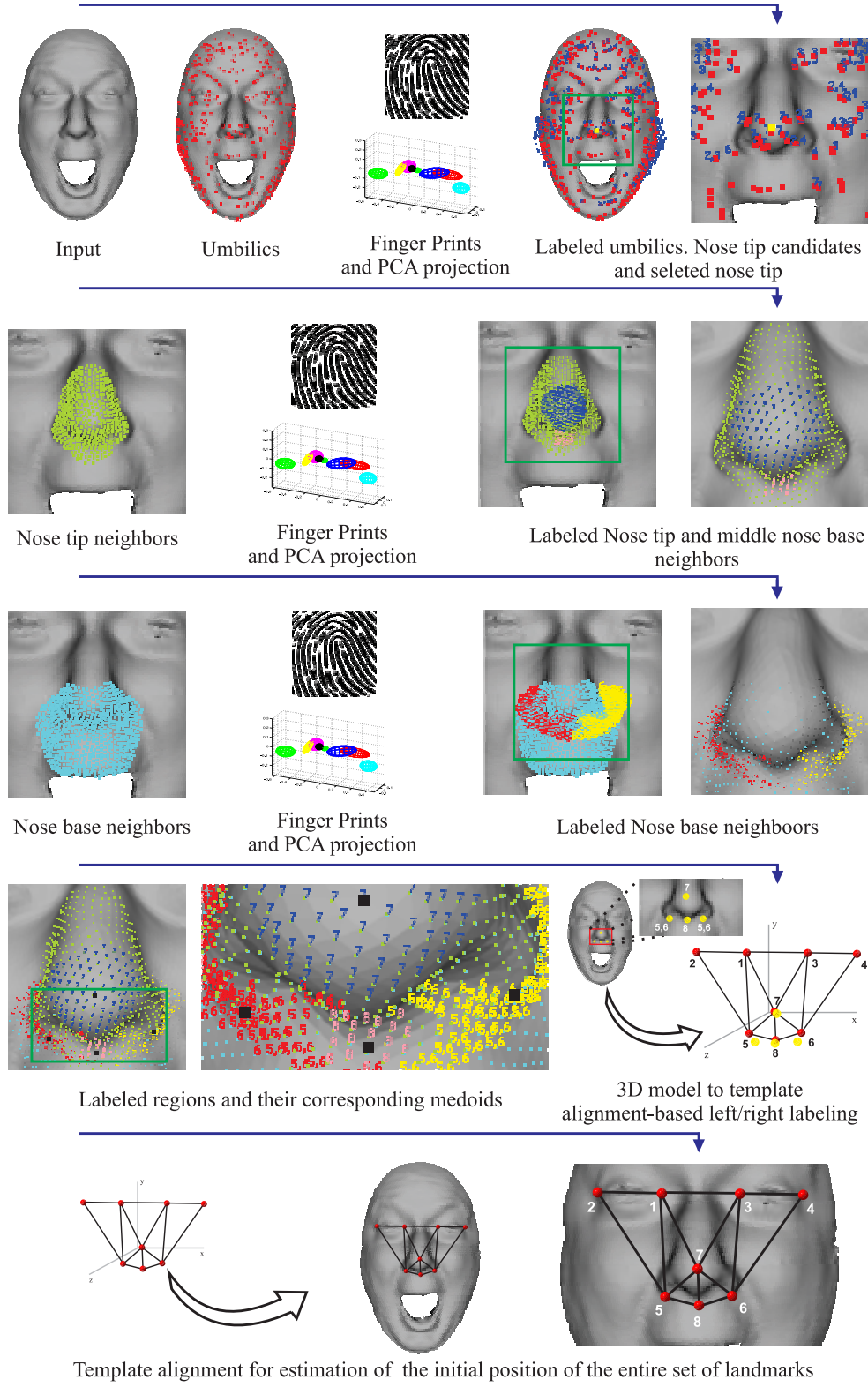


Figure 4: Framework of the proposed initial alignment method.

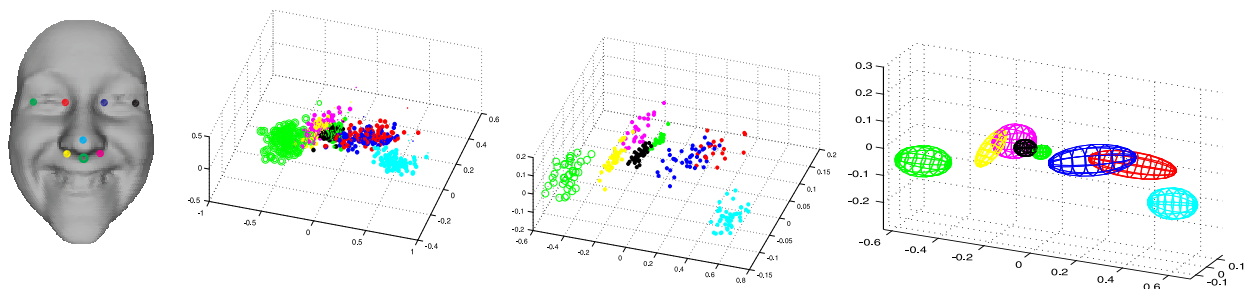


Figure 5: PCA-based clustering. Left to right: Landmarks on a face model. Initial clusters formed with all the samples. Final cluster after removing the samples beyond a one standard deviation from the cluster medoid. Minimum volume enclosing ellipsoids.

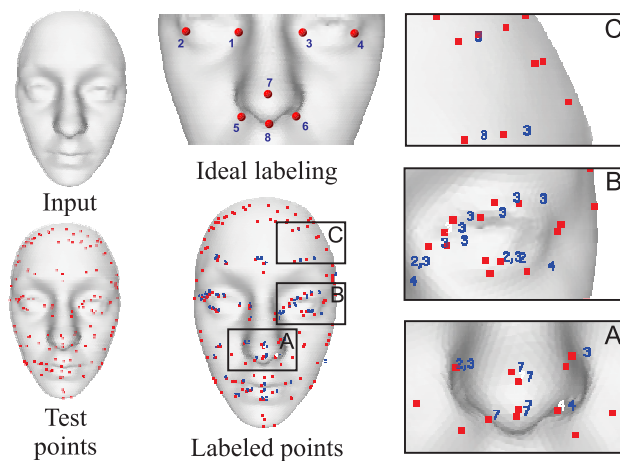


Figure 6: Example of vertices labeling result. (A) Notice how the points on the nose tip region are correctly labeled. (B) Some vertices are assigned to two classes. This situation is because of the left-right symmetry of the features. (C) Points located far from the region of interest are discarded.

which are the points on the surface where the principal curvatures are identical (that is, $k_1 = k_2$). We choose this sampling approach because it can be observed experimentally that most landmark positions are located close to a umbilic, as shown in Fig. 7. For each umbilic the *FP* descriptor is computed, projected into the PCA space, and labeled following the procedure described above.

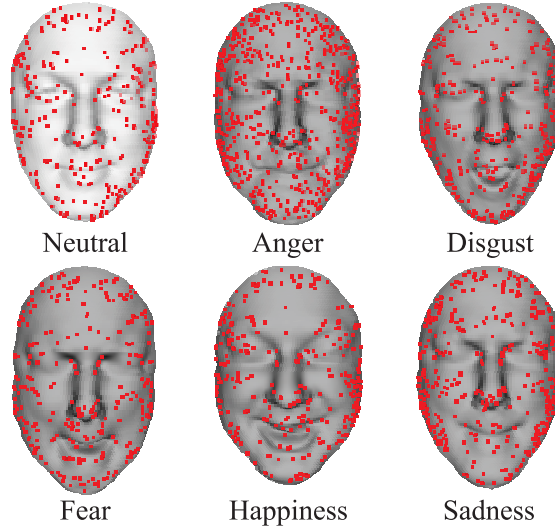


Figure 7: *Umbilics* of different 3D facial models of the same subject performing different expressions. Notice how the *umbilics* are distributed all over the surface, and in most of the cases umbilics are present at the locations of salient facial features.

Once the vertices have been labeled, the next step is to roughly align a template of the upper part of the face with the same structure as the landmark graph structure to the scan (see Fig. 2). The locations of the template vertices are used to define the search space region on which statistical inference is performed. We then perform statistical inference on these search space regions using belief propagation to predict the landmarks. As discussed in Section 3.2.

5 Registration

In this section, we describe how a template is fitted to a 3D scan of the face. The input scan corresponds to a face of a subject performing a facial expression. Fitting a template to this scan is challenging because the facial geometry has large variations due to different face shapes and facial muscle movements. We propose a registration method, where the expression and the shape are fitted separately in order to handle the complexity of the problem. Fig. 8 shows an overview of the proposed method.

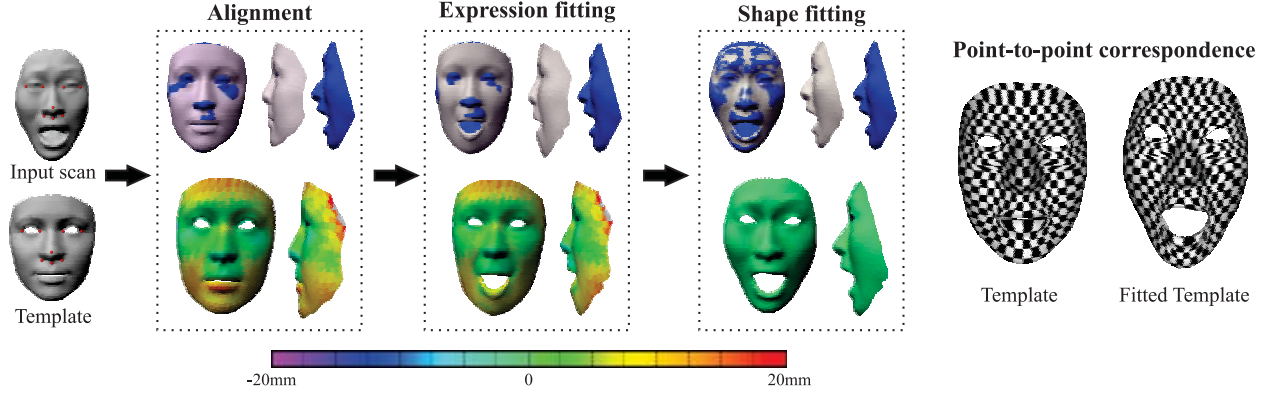


Figure 8: Registration procedure. First, the template and the scan are aligned using the predicted landmarks. Second, the expression is fitted using a blendshape model. Finally, an energy-based surface fitting method is used to fit the shape. At the end, the overlap between the scan and the template is maximized and a point-to-point correspondence for the face shapes in different expressions is obtained.

5.1 Expression Fitting

We address the facial expression fitting problem as a facial rigging problem. In facial rigging, a facial expression is produced by changing a set of parameters associated with the different regions of the face modeled using blendshapes. Conceptually, to generate a facial shape from a 3D rest pose face template, we just move a set of vertices to a new location, e.g., lift an eyebrow or open the mouth (see Fig. 9). In this sense and similar to the approach proposed by Li et al. [9], we model a facial expression as a linear combination of facial blendshapes (denoted by A_i), which are expressed as vectors of displacements from the rest pose (denoted by A_0). Unlike Li et al. where the blendshape model requires the optimization of a weight vector per pose and vertex of the template, we propose a simplified blendshape model where only one weight per pose has to be optimized. The aim of our blendshape model is to catch the pose variations more than the shape variation. Hence, an expression can be generated as

$$P = A_0 + \sum_{i=1}^j \alpha_i A_i, \quad (1)$$

where, A_0 corresponds to the rest pose, $A_i, i > 0$ correspond to the displacements, and α_i ($0 \leq \alpha_i \leq 1$) are the blending weights of the pose P . For each blendshape A_i , Fig. 9 shows the corresponding expressions. This formulation transforms the facial expression fitting problem into an optimization problem, where the value of each α_i has to be estimated.

To solve the fitting problem, the expression template P is aligned to a scan F by setting α_i to 0. Both P and F contain a set of landmarks denoted by \bar{l}_i and l_i , respectively. The landmarks l_i were predicted using the method described in Sections 3 and 4. The alignment is carried out by finding a 3×4 transformation matrix \mathbf{T}_A that minimizes the energy

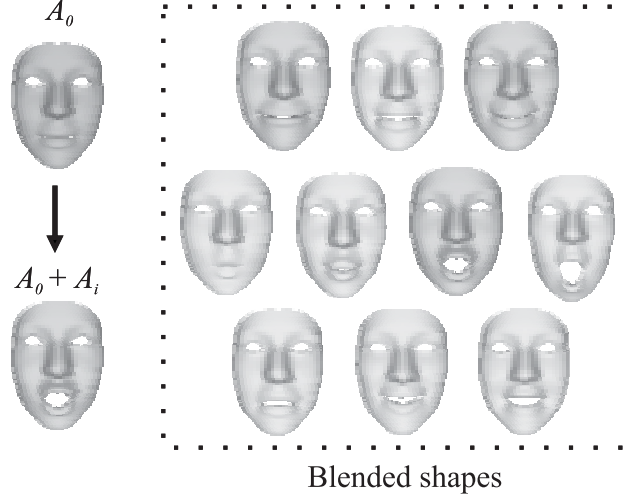


Figure 9: Template: rest pose A_0 and a set of generated blendshapes A_i .

$$E_{lnd} = \sum_{i=1}^L (\mathbf{T}_A \bar{l}_i - l_i),$$

with respect to the 12 parameters in \mathbf{T}_A .

Once P and F are aligned, we find the α_i that best match the expression of F . To achieve this, we divide P into three regions: upper face, chin and mouth (as shown in Fig. 10). The division is motivated by the fact that the chin and lip regions vary drastically from one expression to another (mostly in terms of displacements). Thus it is desirable to inspect the quality of the fitting in each of these regions separately. Face regions like eyebrows and cheeks also change their shapes to produce the expressions but we expect that these changes can be captured during the shape fitting step.

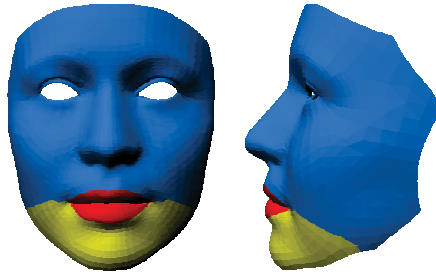


Figure 10: Regions used in the expression fitting procedure.

To fit the expression, we define the energy

$$E_{expression} = \beta E_{NN} + \gamma E_{chin} + \eta E_{mouth}, \quad (2)$$

where,

$$\begin{aligned} E_{NN} &= \sum_r (p_r - NN(p_r))^2, \\ E_{chin} &= \sum_a (p_a - NN(p_a))^2, \\ E_{mouth} &= \sum_b (p_b - NN(p_b))^2, \end{aligned}$$

and β , γ and η are weights, r are the vertices of P , and a and b are the vertices of the chin and mouth region in P , respectively. Here, $NN(p_i)$ indicates the nearest neighbor of a specific vertex p_i . To make the method more robust to both the presence of outliers and misoriented surfaces, we only consider the nearest neighbor in E_{NN} if the angle between the outer normal vectors of the vertex p_i and its nearest neighbor is at most a threshold φ . To force the fit to be exact, the nearest neighbor term for E_{chin} and E_{mouth} is only valid if the angle is at most $\varphi/2$. The expression is fitted by minimizing Eq. 2 with respect to the blending weights α_i . In our experiments we set φ to 80 degrees.

The minimization of $E_{expression}$ is carried out in two stages. In the first stage, we inspect if some movement occurs in the chin; most of the times the displacements are horizontal. Once we know the position of the chin, to refine the match with the expression of the input model, we need to inspect the positions of the lips; in this stage both vertical and horizontal displacements, and shape changes of the lips are matched. Based on this, the expression fitting procedure proceed as follows: First, the weight η is set to 0, thus the minimization is only guided by the E_{NN} and E_{chin} . In this step β is set to 1 and γ is defined as $1 - (V_{chin}/|a|)$, where V_{chin} is the number of valid nearest neighbor in the chin region. The second step begins when $V_{chin} > 0.8|a|$, which means that the overlap between the chin region of P and the model F has reached a good level. At this time, η is set to $1 - (V_{mouth}/|b|)$, where V_{mouth} is the number of valid nearest neighbors in the mouth region. The minimization process ends when $V_{mouth} > 0.6|b|$. This weight variation scheme ensures that the chin and mouth regions of P match the expression of F . The threshold values for V_{chin} and V_{mouth} were choose based on experimental observations.

5.2 Shape Fitting

As most of the changes in terms of movement, especially in the chin and mouth regions, were captured in the expression fitting stage, the next step consists of adapting the shape of the deformed template P to the shape of the scan F . In addition, the changes (displacements) resulting from muscle movement in the eyebrows and forehead are also captured.

The shape fitting is, again, treated as an optimization problem similar to the method proposed by Allen et al. [20] and extended by Li et al. [30]. The goal is to find a set of 3×4 transformation matrices \mathbf{T}_i for each vertex p_i of P such that it is moved to the new location $\mathbf{T}_i p_i$ to fit the shape of F . The transformed version of P is denoted \tilde{P} . The transformation matrices \mathbf{T}_i are obtained by minimizing an energy function, which is a weighted sum of four energy terms.

The first term corresponds to the nearest neighbor term

$$E_{NN} = \sum_{i=1}^r (\mathbf{T}_i p_i - NN(\mathbf{T}_i p_i))^2,$$

where $NN(\mathbf{T}_i p_i)$ indicates the nearest neighbor of a transformed vertex $\tilde{p}_i = \mathbf{T}_i p_i$. This term is only considered if the angle between the outer normal vectors of \tilde{p}_i and its nearest neighbor is at most 80 degrees. The nearest neighbor energy term ensures that the template is deformed to resemble the input scan.

The second energy term corresponds to the smoothness energy

$$E_{smooth} = \sum_i \sum_{j \in R(\tilde{p}_i)} \left(1 - \frac{(\tilde{p}_i - \tilde{p}_j)^2}{(sg)^2} \right) (\mathbf{T}_i - \mathbf{T}_j)^2,$$

where $R(\tilde{p}_j)$ is a set of indices corresponding to points \tilde{p}_j within Euclidean and geodesic distance sg of \tilde{p}_i , with g being the resolution of the mesh and s being a constant. This energy term encourages close-by points to have similar deformations. Note that this term does not encourage a smoothing of the geometry of the mesh. We set $s = 3$ in our implementation.

The third energy term is a regularization term that is conceptually similar to the smoothness energy. This energy term encourages smooth transformations between neighboring vertices of the mesh. We call this energy regularization energy E_{reg} and define it as

$$E_{reg} = \sum_{(i,j) \in E(\tilde{P})} (\mathbf{T}_i - \mathbf{T}_j)^2,$$

where $E(\tilde{P})$ is the set of edges of \tilde{P} . This term prevents adjacent parts of P from being mapped to disparate parts of F , and also encourages similarly-shaped features to be mapped to each other [20].

The final energy term encourages the transformation matrices to be rigid. The rigid energy E_{rigid} , which measures the deviation of the column vectors of \mathbf{T}_i from orthogonality and unit length, is defined as

$$E_{rigid} = \sum_{i=1}^r \left(\left((\mathbf{a}_1^i)^T \mathbf{a}_2^i \right)^2 + \left((\mathbf{a}_1^i)^T \mathbf{a}_3^i \right)^2 + \left((\mathbf{a}_2^i)^T \mathbf{a}_3^i \right)^2 + \right. \\ \left. \left(1 - (\mathbf{a}_1^i)^T \mathbf{a}_1^i \right)^2 + \left(1 - (\mathbf{a}_2^i)^T \mathbf{a}_2^i \right)^2 + \left(1 - (\mathbf{a}_3^i)^T \mathbf{a}_3^i \right)^2 \right),$$

where $\mathbf{a}_1^i, \mathbf{a}_2^i, \mathbf{a}_3^i$ are the first three columns vectors of \mathbf{T}_i .

The energy terms described above are combined in the weighted sum

$$E_{shape} = \lambda_0 E_{NN} + \lambda_1 E_{reg} + \lambda_2 E_{rigid} + \lambda_3 E_{smooth}. \quad (3)$$

The shape is fitted by minimizing E_{shape} with respect to the parameters of \mathbf{T}_i . We start by encouraging smooth and rigid transformations by setting $\lambda_0^0 = 1$, $\lambda_1^0 = 5000$, $\lambda_2^0 = 1000$, and $\lambda_3^0 = 100$. Similar to Li et al. [30], whenever the energy change is negligible, we relax

the weights as $\lambda_1^t = 0.5\lambda_1^{t-1}$, $\lambda_2^t = 0.5\lambda_2^{t-1}$, and $\lambda_3^t = 0.5\lambda_3^{t-1}$ to give more weight to the data term. This allows the template to deform towards the scan. The algorithm iterates until the relative change in energy $(E_{shape}^{i-1} - E_{shape}^i)/E_{shape}^{i-1}$, where i is the iteration number, is less than 0.0001. For each set of weights, we use a quasi-Newton approach [31] to solve the optimization problem, and we perform at most 1000 iterations.

6 Experiments and results

6.1 Database

We use the BU-3DFE [32] database for our experiments. The database consists of 3D face models from 100 subjects (56 Females and 44 Males) in neutral pose and with the following facial expressions: *surprise*, *happiness*, *disgust*, *sadness*, *anger* and *fear*. There are four scans of each facial expression, corresponding to different levels of intensity from *low* to *highest*. As a file containing the raw data of each scan is also available, there are a total of 50 files per subject, 25 raw and 25 corresponding to the cropped faces. Fig. 11 shows snapshots of different scans from the BU-3DFE database.

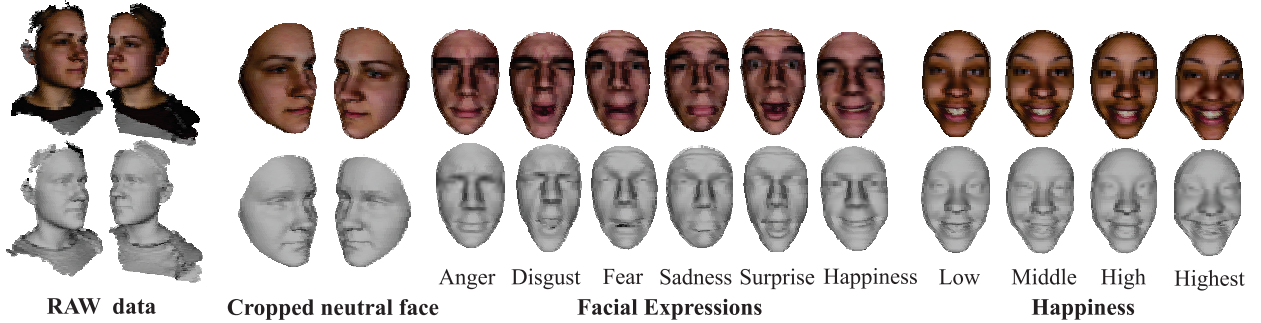


Figure 11: Characteristics of the BU-3DFE database.

6.2 Landmark prediction accuracy

We use two different subsets of models of 50 subjects (25 females and 25 males) to train the landmark prediction model. First, we use a subset T_n consisting of 50 models of the subjects in neutral pose as training set. Second, we use a subset T_e consisting of 350 models of subjects in neutral pose and performing six different facial expressions as training set. As T_n covers the shape variability and T_e covers both shape and expressions variability, we are able to evaluate the importance of the variabilities considered in the training sets.

To evaluate the accuracy of the landmark prediction algorithm, we compute the average error of the distance between a manually located landmark l_i and its corresponding estimation \hat{l}_i . Also, the distribution of the relative error [33] $R_{err} = dist(l_i, \hat{l}_i)/dist_{ref}$ is computed. Here, $dist(l_i, \hat{l}_i)$ is the Euclidean distance between l_i and \hat{l}_i and $dist_{ref}$ is the distance taken as

reference. The value of $dist_{ref}$ is chosen based on the face region that contains the analyzed set of landmarks. To evaluate the accuracy of the location of the landmarks in the region of the eyes, we compute the Euclidean distance between the manually labeled inner and outer corners of the eyes for 80 models with neutral expression. The average of these distances is used as $dist_{ref}$. For the nose region, the procedure is similar, but the distance is computed between the landmarks located at the extremes of the nose base. The distances obtained are 31.973 mm and 27.087 mm , for the eyes and nose regions, respectively.

We evaluate the accuracy of the landmark prediction algorithm over the remaining 50 subjects of the database (31 females and 19 males). The prediction is carried out over 350 models of subjects in both neutral pose and when performing six different facial expressions.

The landmarks were predicted with an error below $dist_{ref}$ in 87% and 95% of the cases for the experiments with T_n and T_e as training databases, respectively. Figs. 12 and 13 show the curves of the relative error distribution obtained using the different training sets. In both experiments, the landmarks located in the nose region are better predicted than the ones located in the eyes region. The average of the relative error curves (see Fig. 14) show the significant improvement in accuracy of the landmark prediction when T_e is used as training set. This indicates that for the configuration of the landmark prediction model used in this work, the variations due to both shape and expression have to be considered.

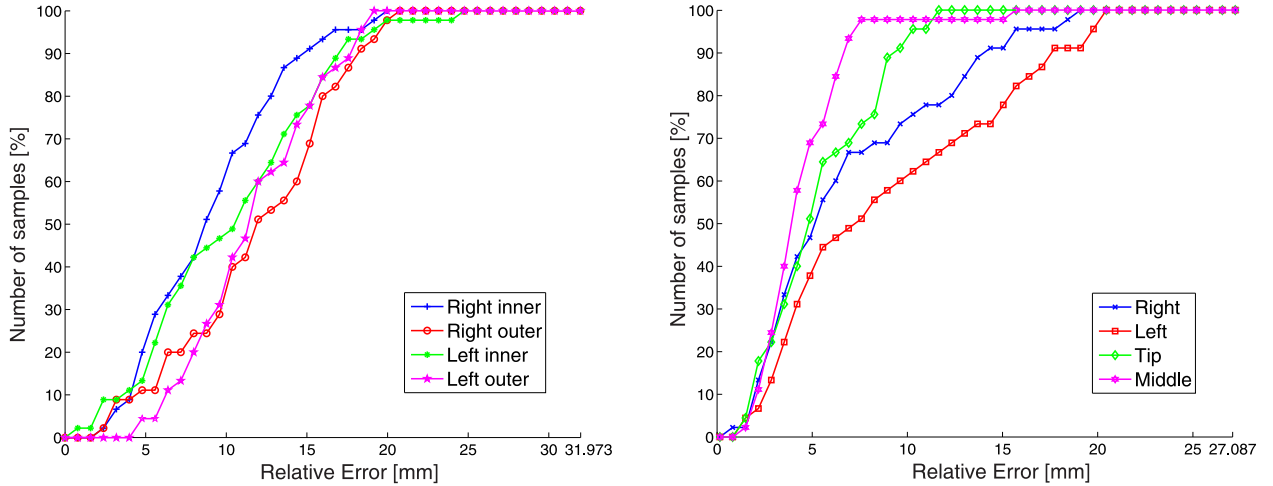


Figure 12: Relative error distribution using data set T_n . Left: Eye landmarks. Right: Nose landmarks.

In addition to the relative error distribution, we compute the average, the standard deviation and the maximum of the error when T_e is used for training (see Table 1). The tip of the nose is predicted with the lowest error and the outer corners of the eyes are predicted with the highest error. One of the reasons that the outer corners of the eyes are not predicted as well as the other landmarks is that the initial position is found based on the alignment of the landmark template (see Fig. 4). This adds an estimation error that is reflected in the high values of the standard deviation.

Although the obtained landmark prediction error appears to be high, it is still possible to

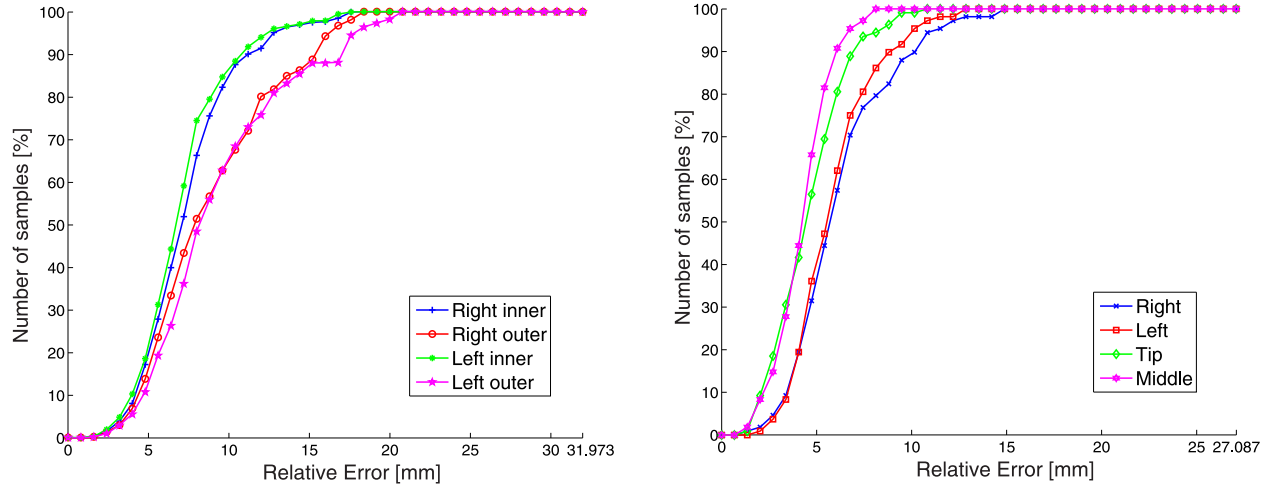


Figure 13: Relative error distribution using data set T_e . Left: Eye landmarks. Right: Nose landmarks.

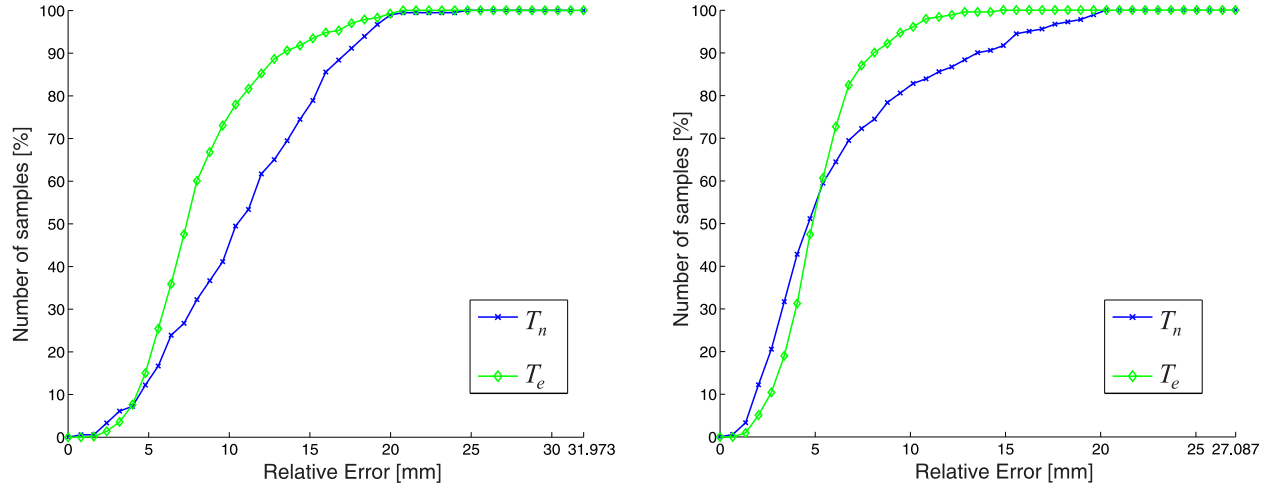


Figure 14: Average of the relative error distribution. Left: Eye landmarks. Right: Nose landmarks.

Landmark	Average \pm Std. deviation [mm]	Max. [mm]
Right inner eye corner	5.46 ± 3.35	17.37
Right outer eye corner	7.88 ± 5.74	27.81
Left inner eye corner	5.01 ± 3.29	19.27
Left outer eye corner	8.01 ± 5.29	22.25
Right subalare	6.16 ± 2.58	14.76
Left subalare	5.80 ± 2.12	12.44
Nose tip	4.57 ± 1.92	10.53
Subnasal	4.20 ± 1.47	8.03

Table 1: Error of landmark prediction with training set T_e .

obtain a proper point-to-point correspondence since the landmarks only provide a guidance for the deformation algorithm. Fig. 15 shows some examples of the landmark prediction results over models of subjects with different facial shapes and performing different expressions. For all the registration experiments, for which results are shown in Section 6.3, we used T_e as training dataset.

6.3 Registration

We tested our dense point-to-point correspondence algorithm on the models where the landmarks were correctly predicted (332 models). To generate the blendshape model we use the blendshapes shown in Fig. 16. Notice that mostly mouth displacements are considered. As the expressions are generated as a linear combination of displacements, to avoid exaggerated undesired expressions, it is important that two blendshapes do not add the same kind of displacement. The third column of Fig. 17 shows examples of the expression fitting results for six different kinds of facial expression. In all cases, the expression of the mouth region of the input model is properly matched after linear blending.

Next we discuss the quality of the results after the final shape fitting step. The fourth column of Fig. 17 shows examples of the shape fitting results. The models are color-coded with respect to the signed distance from the input scan. Note that most points on the models are within $1mm$ of the scan and that the results are visually pleasing. Furthermore, notice how the different expression in the eyebrows are properly fitted. In order to visualize the quality of the correspondences a chess-board texture was applied to the template model (see right of Fig 17). Results of texture transferring show that in most of the face regions, the shape of the deformed template matches the shape of the input model.

We also run tests to verify if the level of the expression affects the quality of the fitting. Fig. 18 depicts how the proposed method is able to correctly fit the template to different

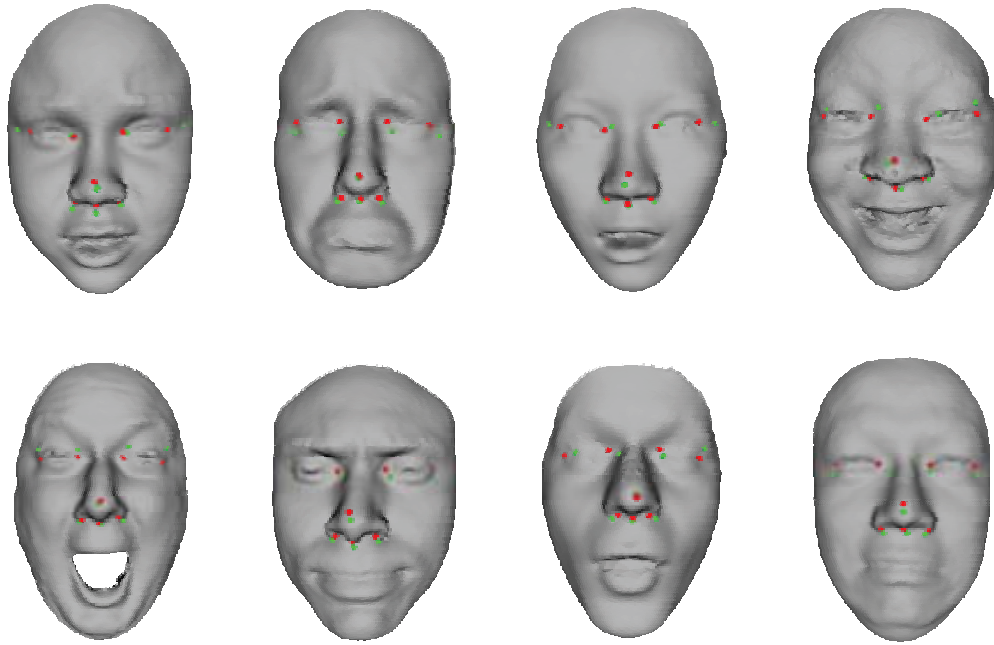


Figure 15: Examples of the landmark prediction results. Red and green spheres correspond to the manually placed and predicted landmarks, respectively. First row: female subjects; Second row: male subjects.

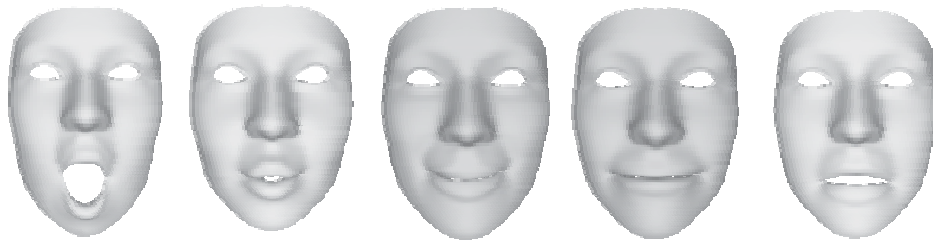


Figure 16: Shapes used to generate the blendshape model.

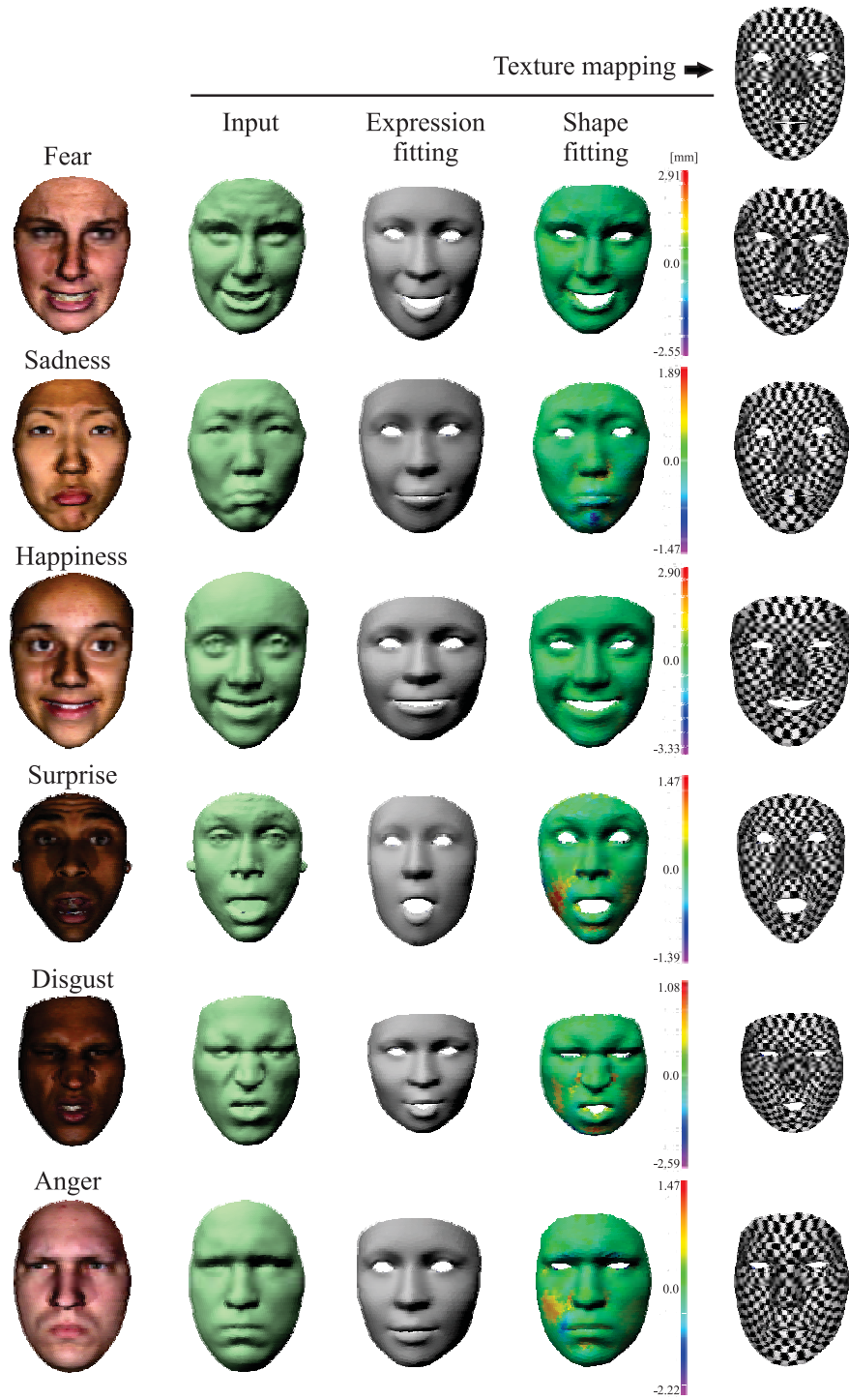


Figure 17: Examples of registration results. The input, fitted expression, error mapped, and texture mapped models are provided for each example.

levels of expressions. For each example, the input, output and textured models are provided. Notice how both slight and pronounced movements of the eyebrows and mouth are properly matched.

Most of the incorrect shape fitting occurs on the inner parts of the lips. As the input scans have information in the area of the teeth, which is not considered in the template model, the algorithm converges to this region, thereby causing miscorrespondences during the shape fitting. Fig. 19 shows an example of the limitations in the shape fitting. Notice how the expression is matched correctly, but the corners of the mouth are not well located, which causes an incorrect fitting on the mouth and chin regions (first row of Fig. 19). The situation becomes critical when the expression is incorrectly matched (second row of Fig. 19). For our experiments, we obtained visually pleasing shape fitting results in 294 (84%) of the tested models.

Additional tests were performed over models with occluded parts. In this case, the template was correctly fitted when the occlusion did not occur in the locations of the landmarks used for the initial alignment. Fig. 20 shows the result of the proposed point-to-point correspondence approach for a model of a subject where the mouth is occluded by a hand. Note that a visually pleasing result is obtained.

7 Conclusions

This paper presented a fully automatic method to compute dense point-to-point correspondences between a set of human face scans with varying expressions. The proposed approach proceeds by learning local shape descriptors and spatial relationships for a set of landmark points. For a new scan, the approach first predicts the landmark points by performing statistical inference on the learned model. The approach then fits a template to the scan in two stages. The first stage fits the expression of the template to the expression of the scan using the predicted landmark points. The second stage fits the shape of the template to the shape of the scan using a non-rigid iterative closest point technique. We applied our approach to 350 models of the BU-3DFE database, and evaluated the results both qualitatively and quantitatively. We showed that for 95% of the models, the landmarks are predicted with an acceptable error, and that for 84% of the models, a visually pleasing correspondence is found. Furthermore, we evaluated the algorithm on a challenging case of a face with occlusion.

The failure cases of the algorithm are mostly caused by noisy data in the mouth area. For future work we plan to design algorithms that can handle this challenging scenario. We will also test the algorithm on a large database of models with different types of occlusion, such as models wearing eyeglasses.

Acknowledgment

This work was supported by the program “*Créditos condonables para estudiantes de Doctorado*” from COLCIENCIAS, by the program “*Convocatoria de apoyo a tesis de posgrado*”

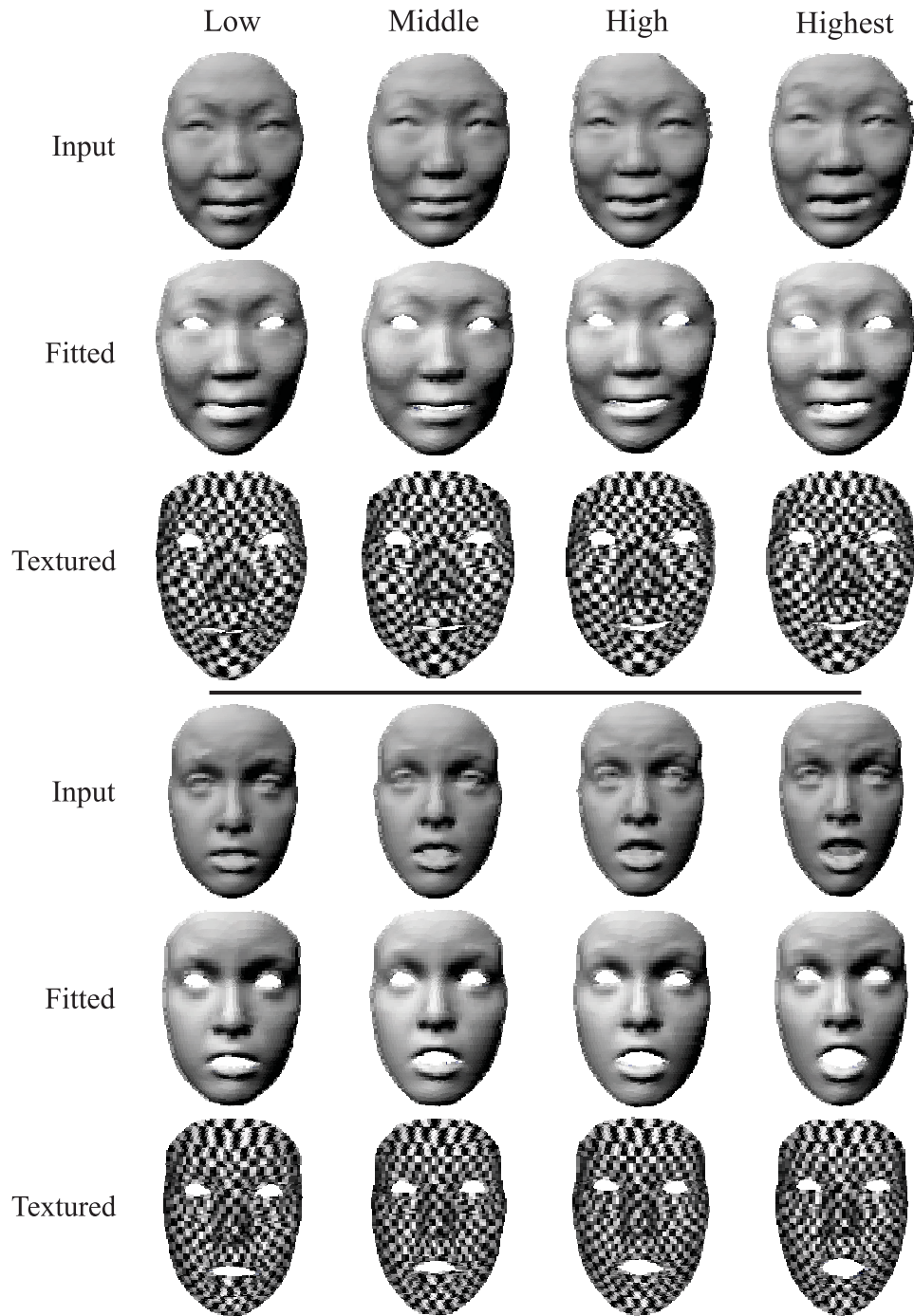


Figure 18: Results of fitting to models of the same subject performing an expression in different levels. Fear (first three rows). Surprise (Last three rows). For each example, first, second, and third rows are the input, output, and textured models, respectively.

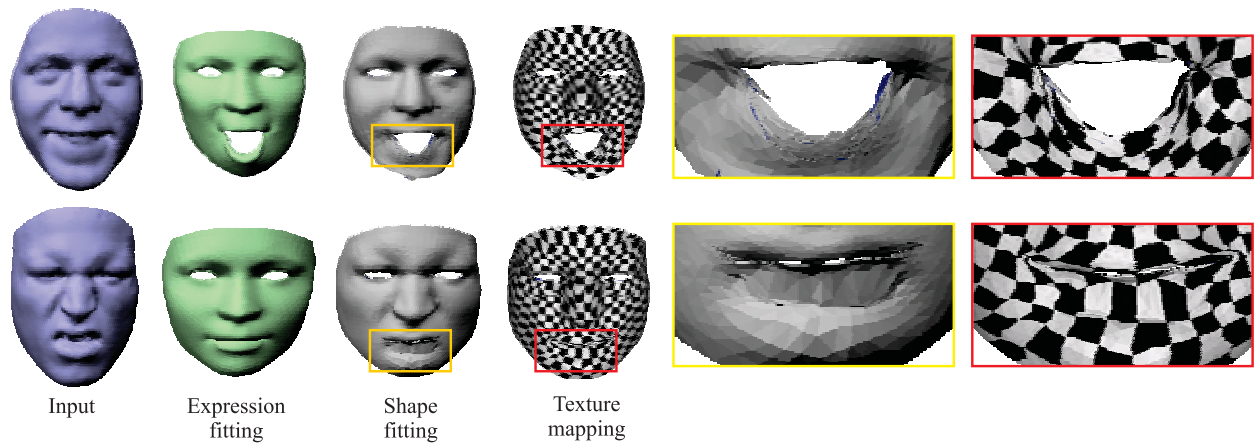


Figure 19: Incorrect shape fitting. The differences in topology of the input and template meshes cause incorrect expression and shape fitting.

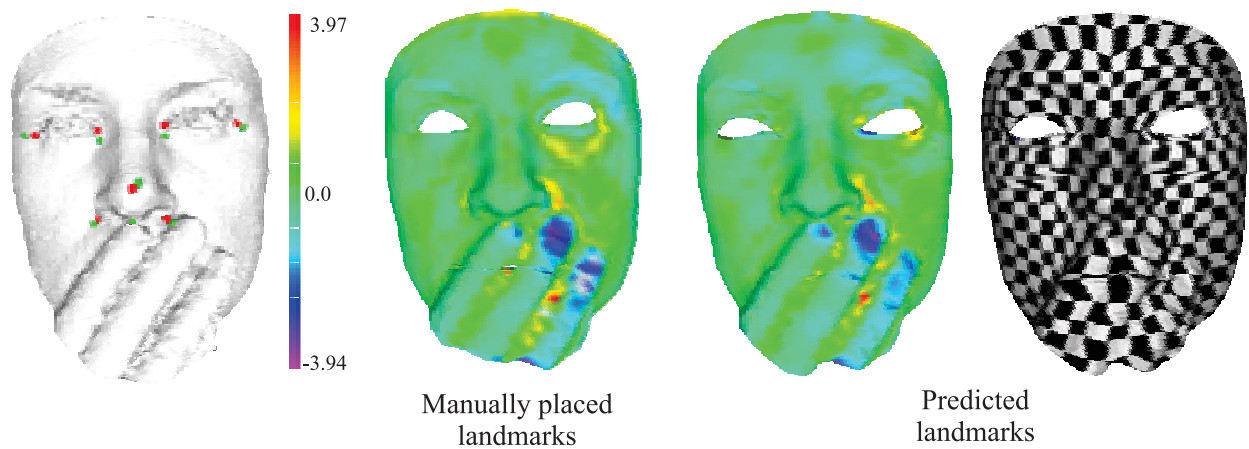


Figure 20: Challenging test scenario. Mapped error models correspond to the fitting result. Test was carried out over one model of the Bosphorus database [34].

- *Doctorados*” from *Dirección de Investigaciones de Manizales*, and by the Cluster of Excellence - Multimodal Computing and Interaction, within the Excellence Initiative of the German Federal Government. We thank Jonathan Boisvert, Alan Brunton, and Pengcheng Xi for helpful discussions.

References

- [1] Ian Dryden and Kanti Mardia. *Statistical Shape Analysis*. Wiley, 2002.
- [2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Conference on Computer Graphics and Interactive Techniques*, pages 187–194, 1999.
- [3] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao. Efficient 3D reconstruction for face recognition. *Pattern Recognition*, 38(6):787–798, 2005.
- [4] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3D morphable model using linear shape and texture error functions. In *IEEE International Conference on Computer Vision*, pages 3–19, 2002.
- [5] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *IEEE International Conference on Computer Vision*, volume 1, pages 59–66, 2003.
- [6] Alan Brunton, Chang Shu, Jochen Lang, and Eric Dubois. Wavelet model-based stereo for fast, robust face reconstruction. In *Canadian Conference on Computer and Robot Vision*, pages 347–354, 2011.
- [7] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Danial Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 3(6):1681–1707, 2011.
- [8] Pengcheng Xi and Chang Shu. Consistent parameterization and statistical analysis of human head scans. *The Visual Computer*, 25(9):863–871, 2009.
- [9] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Transactions on Graphics (SIGGRAPH)*, 29(4):32:1–32:6, 2010.
- [10] S. Mehryar, K. Martin, K.N. Plataniotis, and S. Stergiopoulos. Automatic landmark detection for 3D face image processing. In *IEEE Congress on Evolutionary Computation*, pages 1–7, 2010.
- [11] Zouhour Ben Azouz, Chang Shu, and Mantel Anja. Automatic locating of anthropometric landmarks on 3D human models. In *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 750–757, 2006.
- [12] Clement Creusot, Nick Pears, and Jim Austin. 3D face landmark labelling. In *Proceedings ACM workshop on 3D object retrieval*, pages 27–32, 2010.

- [13] Y. Sun and M. Abidi. Surface matching by 3D point’s fingerprint. In *IEEE International Conference on Computer Vision*, volume 2, pages 263–269, 2001.
- [14] Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.
- [15] G. Passalis, P. Perakis, T. Theoharis, and I. Kakadiaris. Using facial symmetry to handle pose variations in real-world 3D face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1938–1951, 2011.
- [16] I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, Yunliang Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):640–649, 2007.
- [17] Xiaoguang Lu and A. Jain. Deformation modeling for robust 3D face matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1377–1383, 2006.
- [18] Curzio Basso, Pascal Paysan, and Thomas Vetter. Registration of expressions data using a 3D morphable model. In *International Conference on Automatic Face and Gesture Recognition*, pages 205–210, 2006.
- [19] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3D face recognition with a morphable model. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6, 2008.
- [20] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parametrisation from range scans. *ACM Transactions on Graphics (SIGGRAPH)*, 22(3):587–594, 2003.
- [21] Stefanie Wuhler, Chang Shu, and Pengcheng Xi. Landmark-free posture invariant human shape correspondence. *The Visual Computer*, 27(9):843–852, 2011.
- [22] A. Bronstein, M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.
- [23] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16(1):188–197, 2007.
- [24] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (SIGGRAPH)*, 30(4):77:1–77:10, 2011.
- [25] Stefanie Wuhler, Zouhour Ben Azouz, and Chang Shu. Semi-automatic prediction of landmarks on human models in varying poses. In *Canadian Conference on Computer and Robot Vision*, pages 136–142, 2010.

- [26] Trevor Cox and Michael Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall CRC, 2001.
- [27] J. Yedidia, W. Freeman, and Y. Weiss. *Understanding Belief Propagation and Its Generalizations*. Science & Technology Books, 2003.
- [28] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers, 2006.
- [29] Frederic Cazals and Marc Pouget. Smooth surfaces, umbilics, lines of curvatures, foliations, ridges and the medial axis: a concise overview. Technical Report RR-5138, INRIA, 2004.
- [30] Hao Li, Bart Adams, J. Guibas, Leonidas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 28(5):175:1–175:10, 2009.
- [31] Dong Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
- [32] L. Yin, X. Wei, J. Wang, Y. Sun, and M. Rosato. A 3D facial expression database for facial behavior research. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 211–216, 2006.
- [33] Oliver Jesorsky, Klaus Kirchberg, and Robert Frischholz. Robust face detection using the hausdorff distance. In *International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 90–95, 2001.
- [34] Arman Savran, Necse Alyuz, Hamdi Dibekliouglu, Oya Celiktutan, Berk Gokberk, Bulent Sankur, and Lale Akarun. Bosphorus database for 3D face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56, 2008.